UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/671,844	09/25/2003	**Sujat Jamil**	MP1491	8695

26703 7590 07/01/2009

HARNESS, DICKEY & PIERCE P.L.C.
5445 CORPORATE DRIVE
SUITE 200
TROY, MI 48098

EXAMINER	
FENNEMA, ROBERT E	
ART UNIT	**PAPER NUMBER**
2183	

MAIL DATE	DELIVERY MODE
07/01/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/671,844	JAMIL ET AL.	
	Examiner	Art Unit	
	ROBERT E. FENNEMA	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 April 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-41 have been considered. Claims 1, 4-7, 9, 11, 14-17, and 21-32 amended as per Applicant's request. Claims 39-41 added as per Applicant's request.

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 4/30/2009 has been entered.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4. Claims 1-6, 8-9, 11-16, 18-19 and 31-38 are rejected under 35 U.S.C. 102(b) as being anticipated by Merchant et al. (US Patent 6,385,715, herein Merchant).

5. As per claim 1, Merchant teaches a method comprising:

issuing an instruction selected from a queue (Column 3, Lines 25-33; Column 3, Lines 43-47);

enqueuing the instruction issued within a recirculation queue (Column 8, Lines 13-16);

selectively setting a state of the instruction in the recirculation queue to one of a blocked state if completion of the instruction is prevented by a first detected blocking condition and an unblocked state if completion of the instruction is prevented by a second detected blocking condition (Column 8, Lines 42-53, the long latency instruction is a blocked instruction, the dependents are not blocked, as they are not directly dependent upon the long-latency event (such as a cache miss), but there is still a blocking condition which exists (the independent/long-latency instruction being blocked is blocking them). A state is set, as seen in Column 9, Lines 28-36, as the replay queue knows what the blocking conditions are for the long-latency instructions, the state must be set for that to be recognized),

wherein the first detected blocking condition corresponds to a first data hazard (Column 7, Lines 64-67, a long-latency event), and

wherein the second detected blocking condition corresponds to a second data hazard (Column 8, Lines 42-50, the long latency instruction is the blocking condition); and

reissuing the instruction from the recirculation queue if a third detected blocking condition of at least one instruction within the recirculation queue, other than the

instruction, is satisfied (Column 8, Lines 42-53. Also see Column 11, Lines 19-22, when one instruction can go in the queue, all instructions may be unloaded).

6. As per claim 2, Merchant teaches: The method of claim 1, wherein issuing comprises:

arbitrating between a plurality of queues to select a queue (Column 9, Lines 42-52);

selecting a current instruction from the queue selected (Column 9, Lines 42-52); and

issuing the current instruction for the queue selected (Column 9, Lines 42-52).

7. As per claim 3, Merchant teaches the method of claim 2, wherein issuing the current instruction comprises:

determining a state of the current instruction (Column 9, Lines 58-64);

selecting an alternate queue from the plurality of queues if the state of the instruction is blocked (Column 9, Lines 65-67); and

issuing an instruction selected from the alternate selected queue (Column 9, Lines 42-55).

8. As per claim 4, Merchant teaches the method of claim 1, wherein enqueuing comprises:

detecting a blocking condition that corresponds to the first detected blocking

condition prohibiting the instruction issued from completion (Column 8, Lines 54-67);

placing the instruction within the recirculation queue (Column 9, Lines 1-8);

setting the state of the instruction as blocked to prohibit reissue of the instruction (Column 9, Lines 25-33) (All instructions in the replay queue are blocked and will not be not be reissued until the blocking condition has been cleared.); and

storing the first detected blocking condition (Column 12, Lines 51-57) (The fact that the replay unloading controller can selectively choose which long latency instruction is referenced by data return signal shows that the blocking condition was stored.).

9. As per claim 5, Merchant teaches the method of claim 1, further comprising:

identifying blocking conditions of instructions within the recirculation queue (Column 12, Lines 51-57);

determining whether any blocking condition of any instruction within the recirculation queue is satisfied (Column 12, Lines 51-57); and

enabling recirculation of instructions from the recirculation queue by setting the state of each instruction within the recirculation queue to the unblocked state if any blocking condition is satisfied (Column 12, Lines 58-60).

10. As per claim 6, Merchant teaches the method of claim 1, wherein reissuing instructions comprises:

receiving a request to issue an instruction contained within the recirculation

queue (Column 12, Lines 51-55) (The data return signal is a request to issue since instructions are issued based on the receiving of data.);

determining a state of a current instruction of the recirculation queue (Column 12, Lines 55-57);

issuing the current instruction if the state of the current instruction is the unblocked state in response to the received request (Column 12, Lines 57-60; Column 12, Lines 14-21); and

disregarding the request if the state of the current instruction is the blocked state (Column 12, Lines 57-60; Column 12, Lines 14-21) (The unloading controller chooses which of the replay queues should be unloaded based on the data return signal based on the control signals to the mux, the instruction is either issued if it was the instruction chosen by the unloading controller or denied if it was not chosen.).

11. As per claim 8, Merchant teaches the method of claim 1, wherein reissuing the instructions comprises:

issuing an unblocked instruction in response to a received request (Column 9, Lines 28-36},

enqueuing the reissued instruction if a blocking condition of the instruction remains unsatisfied (Column 7, Lines 9-12);

setting the state of the reissued instruction to the blocked state (Column 8, Lines 42- 53); and

storing the blocking condition (Column 12, Lines 51-57. The fact that the replay

unloading controller can selectively choose which long latency instruction is referenced by data return signal shows that the blocking condition was stored.).

12. As per claim 9, Merchant teaches the method of claim 1, wherein one of the first and second detected blocking conditions is one of a data blocking condition and a resource blocking condition (Column 8, Lines 13-16).

13. Claims 11-16 and 18-19 are substantially similar to Claims 1-6 and 8-9, and are rejected for the same reasons.

14. As per Claim 31, Merchant teaches: A method comprising:

Issuing an instruction selected from a queue (Column 3, Lines 25-33; Column 3, Lines 43-47);

enqueuing the instruction issued within a recirculation queue selectively setting a state of the instructions in the recirculation queue to one of a blocked state if completion of the instruction is prevented by a first detected blocking condition and an unblocked state if completion of the instruction is prevented by a second detected blocking condition (Column 8, Lines 42-53, the long latency instruction is a blocked instruction, the dependents are not blocked, as they are not directly dependent upon the long-latency event (such as a cache miss). A state is set, as seen in Column 9, Lines 28-36, as the replay queue knows what the blocking conditions are for the long-latency instructions, the state must be set for that to be recognized),

wherein the first detected blocking condition corresponds to a first data hazard (Column 7, Lines 64-67, a long-latency event), and

wherein the second detected blocking condition corresponds to a second data hazard (Column 8, Lines 42-50, the long latency instruction is the blocking condition);

resetting the state of the instruction within the recirculation queue if a third detected blocking condition of at least one instruction within the recirculation queue, other than the instruction, is satisfied (Column 8, Lines 42-53); and

reissuing the instruction from the recirculation queue if a state of the instruction is indicated as the unblocked state (Column 8, Lines 42-53).

15. As per Claim 32, Merchant teaches: A method comprising:

issuing a first instruction from a queue (Column 3, Lines 25-33 and 43-47);

detecting a first blocking condition for the first instruction prior to execution of the first instruction (Column 8, Lines 12-18, also see Column 5, Lines 42-49 for other blocking conditions, such as lack of source data or waiting for memory);

setting the first instruction to one of a blocked state when the first blocking condition corresponds to a first data hazard and an unblocked state when the first blocking condition corresponds to a second data hazard (Column 8, Lines 12-18, it is blocked until the condition is cleared, or see Column 7, Lines 9-13 for unblocked instruction cases);

enqueuing the first instruction within a recirculation queue selectively setting a state of the instruction in the recirculation queue to one of the blocked state and the

unblocked state if completion of the instruction is prevented by the first blocking condition (Column 8, Lines 42-53, the long latency instruction is a blocked instruction, the dependents are not blocked, as they are not directly dependent upon the long-latency event (such as a cache miss). A state is set, as seen in Column 9, Lines 28-36, as the replay queue knows what the blocking conditions are for the long-latency instructions, the state must be set for that to be recognized); and

reissuing the first instruction from the recirculation queue if the first blocking condition is satisfied (Column 8, Lines 16-18).

16. As per Claim 33, Merchant teaches: The method of claim 32 further comprising:

detecting a second blocking condition for a second instruction, wherein the second blocking condition differs from the first blocking condition and the second instruction differs from the first instruction (Column 8, Lines 12-18, if it can detect one instructions blocking condition, it can detect other instructions blocking conditions as well); and

reissuing the first instruction from the recirculation queue if the second blocking condition is satisfied (Column 11, Lines 19-21, when the "first" instruction in the queue clears its blocking condition (which could be the "second" or "first" instruction in terms of the claim language), all instructions re-issue).

17. As per Claim 34, Merchant teaches: The method of claim 32 further comprising:

setting the first instruction to the unblocked state based on the first blocking condition (Column 7, Lines 8-12); and

enqueuing the first instruction within the recirculation queue in the unblocked state until the first blocking condition is satisfied (Column 7, Lines 8-12, it will sit in the unblocked state in the queue until it can properly issue).

18. As per Claim 35, Merchant teaches: The method of claim 34 further comprising:

detecting a second blocking condition for a second instruction, wherein the second blocking condition differs from the first blocking condition and the second instruction differs from the first instruction (Column 8, Lines 12-18, if it can detect one instructions blocking condition, it can detect other instructions blocking conditions as well);

setting the second instruction to the blocked state based on the second blocking condition (Column 8, Lines 12-18, it is put in the replay queue); and

enqueuing the second instruction within the recirculation queue in the blocked state until the second blocking condition is satisfied (Column 8, Lines 16-18).

19. As per Claim 36, Merchant teaches: The method of claim 32, wherein enqueuing comprises:

determining whether the first blocking condition is a transient blocking condition (Column 7, Lines 8-12 and Column 8, Lines 12-18, it determines if it is a short or long latency event); and

setting the first instruction to the unblocked state if the first blocking condition is transient (Column 7, Lines 8-12).

20. As per Claim 37, Merchant teaches: The method of claim 1, wherein selectively setting the state of the instruction in the recirculation queue to the unblocked state is based on whether the detected blocking condition is a transient blocking condition (Column 8, Lines 12-18, the system differentiates between long latency instructions, and the dependent instructions, which are "transient" in the sense that they aren't long latency (just dependent on one)).

21. As per Claim 38, Merchant teaches: The method of claim 37, further comprising delaying reissue of the instruction from the recirculation queue when the instruction is in the unblocked state (Column 11, Lines 19-26, it is delayed because the instruction upon which it depends is still not ready).

Claim Rejections - 35 USC § 103

22. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

23. Claims 10 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Merchant, in view of Official Notice.

24. As per claims 10 and 20: Merchant et al. do not explicitly disclose using a circular queue. However, they do disclose using a FIFO queue (Merchant et al.: Column 9, Lines 33-36). Using a circular FIFO queue is well-known in the art since it is easier to use a circular FIFO queue than shifting each entry after each dequeue (Official Notice). Additionally, Examiner notes that the KSR decision has indicated that the claim is obvious if it would have been "obvious to try" from a finite number of predictable solutions. There are only a handful of ways to design a queue, with a circular queue being one of them, therefore, it cannot be a patentable distinction to simply make a queue circular, as opposed to any other type of well-known queue type.

25. Claims 7, 17, 21-30, and 39-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Merchant.

26. As per claim 7, Merchant teaches the method of claim 1, wherein enqueuing comprises:

determining whether the detected blocking condition preventing the instruction issued from completion is a transient blocking condition (Column 8, Lines 12-18, the system differentiates between long latency instructions, and the dependent instructions, which are "transient" in the sense that they aren't long latency (just dependent on one));

setting a state of the instruction to an unblocked state (Column 11, Lines 19-26, only the "agent" instructions are considered to be blocked); and

resetting a state of each instruction within the recirculation queue to an unblocked state (Column 9, Lines 28-36), but fails to teach:

if the detected blocking condition is transient.

Merchant teaches that instructions can be in either a blocked or unblocked state, however, a transient instruction in Merchant would typically move to the replay loop, and not the replay queue. However, *In re Lindberg, 93 USPQ 23 (CCPA 1952)* has established that making two parts integral are not a patentable distinction. Examiner recognizes that Merchant describes several advantages for having the two be separate, as described in Column 8. However, Examiner notes that one of ordinary skill in the art would also recognize that the separation of logic creates additional complexity, and if one was willing to make the sacrifice of performance in order to reduce complexity, one of ordinary skill in the art would remove the replay loop and send everything to the replay queue. Therefore, if the queue was to function as both the replay queue and the replay loop, transient instructions would also be sent to the queue.

Claim 17 is substantially similar to Claim 7 and is rejected for the same reasons.

27. As per claim 21, Merchant teaches: An apparatus, comprising:

a received instruction queue to store received instructions (Column 3, Lines 25-33; Column 3, Lines 43-47);

a recirculation queue (Figure 1, the combination of loop 156 and queue 170,

starting at controller 154); arbitration logic to select one of the received instruction queue and the recirculation queue from which to issue a current instruction (Column 9, Lines 42-52); and

blocked instruction detection logic to identify instructions blocked from execution by detected blocking conditions, and to enqueue the instructions onto the recirculation queue in one of a blocked state and an unblocked state, including a respective blocking condition of each instruction within the recirculation queue, wherein instructions are enqueued onto the recirculation queue in the unblocked state (Column 9, Lines 1-8), and

wherein the transient blocking condition corresponds to a data hazard (Column 7, Lines 9-12), but fails to teach:

having a transient blocking condition enqueued in the queue.

Merchant teaches that instructions can be in either a blocked (non-transient) or unblocked (transient) state, however, a transient instruction in Merchant would typically move to the replay loop, and not the replay queue. However, *In re Lindberg, 93 USPQ 23 (CCPA 1952)* has established that making two parts integral are not a patentable distinction. Examiner recognizes that Merchant describes several advantages for having the two be separate, as described in Column 8. However, Examiner notes that one of ordinary skill in the art would also recognize that the separation of logic creates additional complexity, and if one was willing to make the sacrifice of performance in order to reduce complexity, one of ordinary skill in the art would remove the replay loop and send everything to the replay queue. Therefore, if the queue was to function as

both the replay queue and the replay loop, transient instructions would also be sent to the queue.

28. As per claim 22, Merchant teaches: The apparatus of claim 21, wherein the blocked instruction detect logic further comprises:

blocked condition satisfaction logic to:

detect whether a blocking condition of an instruction within the recirculation queue is satisfied, and

set a state of each instruction within the recirculation queue to the unblocked state if the blocking condition of the instruction within the recirculation queue is satisfied (Column 9, Lines 25-36).

29. As per claim 23, Merchant teaches: The apparatus of claim 21, wherein the arbitration logic to:

determine a state of a selected instruction,

select the received instruction queue if the state of the selected instruction is blocked, and

issue an instruction selected from the received instruction queue (Column 9, Lines 64-67).

30. As per claim 24, Merchant teaches: The apparatus of claim 21, wherein the blocked instruction detect logic to:

determine whether the detected blocking condition is a transient blocking condition (Column 8, Lines 12-18, the system differentiates between long latency instructions, and the dependent instructions, which are "transient" in the sense that they aren't long latency (just dependent on one)),

set a state of the instruction placed within the queue to the unblocked state if the detected blocking condition is transient (Column 11, Lines 19-26, only the “agent” instructions are considered to be blocked), and

reset a state of each instruction within the recirculation queue to the unblocked state to enable reissue of instructions contained within the recirculation queue (Column 9, Lines 28-36).

31. As per claim 25, Merchant teaches: The apparatus of claim 21, wherein the blocked instruction detect logic to:

enqueue a reissued instruction if a blocking condition of the instruction remains unsatisfied (Column 7, Lines 9-12),

set a state of the reissued instruction to the blocked state (Column 8, Lines 42-53) and

store the blocking condition (Column 12, Lines 51-57) (The fact that the replay unloading controller can selectively choose which long latency instruction is referenced by data return signal shows that the blocking condition was stored.).

32. As per claim 26, Merchant teaches: A system comprising:

a memory controller coupled to a memory (Column 4, Lines 20-23);

a processor coupled to the memory via a bus (Figure 1, item 100), the processor including:

a bus interface unit coupling an execution core to a cache memory including:

a received instruction queue to store received instructions (Column 3, Lines 25-33),

a recirculation queue (Figure 1, the combination of loop 156 and queue 170, starting at controller 154), arbitration logic to select one of the received instruction queue and the recirculation queue from which to issue a current instruction (Column 9, Lines 42- 52), and

blocked instruction detection logic to identify instructions blocked from execution by detected blocking conditions, and to enqueue the instructions onto the recirculation queue by selectively setting states of the instructions in the recirculation queue to one of a blocked state and an unblocked state, including a respective blocking condition of each instruction within the recirculation queue (Column 8, Lines 42-53, the long latency instruction is a blocked instruction, the dependents are not blocked, as they are not directly dependent upon the long-latency event (such as a cache miss). A state is set, as seen in Column 9, Lines 28-36, as the replay queue knows what the blocking conditions are for the long-latency instructions, the state must be set for that to be recognized), and

wherein the transient blocking condition corresponds to a data hazard (Column 7, Lines 9-12), but fails to teach:

wherein instructions having a transient blocking condition are enqueued onto the recirculation queue in the unblocked state.

Merchant teaches that instructions can be in either a blocked or unblocked state, however, a transient instruction in Merchant would typically move to the replay loop, and not the replay queue. However, *In re Lindberg, 93 USPQ 23 (CCPA 1952)* has established that making two parts integral are not a patentable distinction. Examiner recognizes that Merchant describes several advantages for having the two be separate, as described in Column 8. However, Examiner notes that one of ordinary skill in the art would also recognize that the separation of logic creates additional complexity, and if one was willing to make the sacrifice of performance in order to reduce complexity, one of ordinary skill in the art would remove the replay loop and send everything to the replay queue. Therefore, if the queue was to function as both the replay queue and the replay loop, transient instructions would also be sent to the queue.

33. Claims 27-30 recite the same limitations as claims 22-25 and are rejected for the same reasons.

34. As per Claim 39, Merchant teaches: The method of claim 1,

wherein the first detected blocking condition corresponds to a non-transient blocking condition based on the first data hazard (Column 7, Lines 64-67, a long latency event),

wherein the second detected blocking condition corresponds to a transient blocking condition based on the second data hazard (Column 7, Lines 8-13), and

wherein the first data hazard is different than the second data hazard (One is long latency, one is short), but fails to teach:

an instruction with a transient blocking condition going to the queue.

Merchant teaches that instructions can be in either a blocked or unblocked state, however, a transient instruction in Merchant would typically move to the replay loop, and not the replay queue. However, *In re Lindberg, 93 USPQ 23 (CCPA 1952)* has established that making two parts integral are not a patentable distinction. Examiner recognizes that Merchant describes several advantages for having the two be separate, as described in Column 8. However, Examiner notes that one of ordinary skill in the art would also recognize that the separation of logic creates additional complexity, and if one was willing to make the sacrifice of performance in order to reduce complexity, one of ordinary skill in the art would remove the replay loop and send everything to the replay queue. Therefore, if the queue was to function as both the replay queue and the replay loop, transient instructions would also be sent to the queue.

35. As per Claim 40, Merchant teaches: The method of claim 39, wherein the second data hazard is based on a plurality of instructions that include the instruction, wherein

the plurality of instructions are issued to a single address (Column 5, Line 41, it's not a long-latency fault, but would create improper execution, thus would be classified as a transient blocking condition).

36. As per Claim 41, Merchant teaches: The method of claim 39, wherein the first data hazard is based on at least one of memory latency and bandwidth (Column 8, Lines 12-18, an access to main memory is a latency issue).

Response to Arguments

37. Regarding Applicant's arguments on Page 22, essentially that Merchant does not teach that the dependent instructions are blocked by blocking conditions that correspond to data hazards, Examiner notes that in the Applicant's specification, a data hazard is considered to be "where a subsequent instruction needs to see the side effects of a previous instruction in order to execute". Because a dependent instruction needs to see the result of the independent instruction in order to execute, Examiner believes that it falls under the category of a data hazard.

However, in considering this argument, the Examiner also considered that although the dependent instructions are blocked due to data hazards, it might not be fair to consider them to be "transient" data hazards, if they depend upon a non-transient data hazard. Even though they themselves are not directly blocked from the non-transient hazard, they are indirectly blocked from it, and would take longer to execute than a non-transient hazard instruction. Thus, Examiner has provided a new grounds of

rejection for the claims which indicate transient hazards, as Examiner believes that it is only appropriate at this point to consider transient hazard instructions to be the instructions which appear in Merchant's replay loop. However, Examiner believes that the act of taking two separate paths, and combining them into one, is not necessarily a patentable distinction. While the separate paths do have significant advantages, the separate paths also necessitate extra hardware, and thus one may be willing to sacrifice the performance in order to achieve simplicity.

However, if the Applicant can show to the Examiner that this would not be the case, then the claims would likely be in condition for allowance, as Examiner believes Merchant is the best and closest art, and that other single-queue systems do not contain the other limitations present in the claims. If Applicant believes an interview would be helpful, they are welcome to contact the Examiner at the number below.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ROBERT E. FENNEMA whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Thursday, 9:30-6:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Eddie P Chan/
Supervisory Patent Examiner, Art Unit 2183

Robert E Fennema
Examiner
Art Unit 2183

RF